

ArQVIA: Prototipo WebApp de VSLAM para Reconstrucciones Virtuales 3D en Espacios Arqueológicos

Explorando la Arqueología con Técnicas Avanzadas multimedia, CBIR, deconstrucción Fotogramétrica e Inteligencia Artificial

The logo of the Universitat Oberta de Catalunya (UOC), consisting of the letters 'UOC' in a stylized, bold, blue font.

Autor: Alicia Colmenero Fernández

Tutor: Daniel Julià Lundgren

Profesor/a: Carlos Casado Martínez.

TFG Multimedia

Memoria Final ArQVIA

Grado Multimedia

20.662 Desarrollo de aplicaciones interactivas

Universitat Oberta
de Catalunya

19/06/2023

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	ArQVIA prototipo <i>mobile app</i> para VSLAM de Reconstrucciones Virtuales 3D en espacios arqueológicos con <i>reverse image search</i> .
Nombre del autor:	Alicia Colmenero Fernandez
Nombre del consultor/a:	Daniel Julià Lundgren
Nombre del PRA:	Carlos Casado Martínez
Fecha de entrega:	19/06/2023
Titulación o programa:	Grado Multimedia
Área del Trabajo Final:	Desarrollo de aplicaciones interactivas
Idioma del trabajo:	Castellano
Palabras clave	Machine learning, VSlam, Archaeology app

Resumen del Trabajo

El logro de este trabajo es el desarrollo de un prototipo de **mobile web app** que implementa una **metodología novedosa para la visualización interactiva** de reconstrucciones virtuales 3D localizadas en espacios arqueológicos con inteligencia artificial.

La arqueología virtual permite la visualización de contenidos que ayudan al visitante a comprender el yacimiento a través de la reconstrucción tridimensional del enclave original. Uno de los problemas principales para la interacción en la visualización es el denominado VSLAM (*visual simultaneous localization and mapping*). El problema fundamental reside en obtener una coincidencia precisa o "matching" entre nuestra posición real y la réplica virtual a escala. Las distintas técnicas existentes: reconocimiento de marcadores; análisis, extracción y comparativa de características de la imagen de entrada contra fotogramas clave o nube de puntos, presentan problemas de portabilidad, instalación software, altos costes computacionales (sistemas embebidos, GPUs) o compilación de librerías para la realización de cálculos.

El método propuesto permite el reconocimiento desde una posición libre **realizando consulta de imágenes mediante ejemplos (CBIR) Content-Based Image Retrieval, recuperación de imágenes basada en contenido, que se ejecuta en un servidor sobre su VRAM, de manera que el dispositivo solamente realiza el envío de una imagen y gestiona la respuesta de la web usando el giroscopio con un método intuitivo y natural**. A través de la extracción de características, se compara una toma realizada in situ RGB con las características aprendidas de los renders realizados sobre un modelo sintético preciso del emplazamiento. Una vez encontrada la coincidencia y realizadas unas operaciones con el resultado el servidor devolverá una panorámica 360x360 con la reconstrucción de la cámara más próxima almacenada en un repositorio y adaptando la vista al grado de rotación.

ArQVIA ejecuta la consulta en modo request-response contra el backend (Python, JavaScript), devolviendo al visitante la visualización tridimensional manejable por el giroscopio como "magic window". Una mobile app accesible desde cualquier punto del yacimiento documentado virtualmente, ejecutable desde un dispositivo móvil monocular sin prestaciones especiales.

Abstract

The achievement of this work is the development of a **prototype mobile web** app that implements a **novel methodology for the interactive visualization** of 3D virtual reconstructions located in archaeological spaces using artificial intelligence.

Virtual archaeology enables the visualization of content that helps visitors understand the archaeological site through the three-dimensional reconstruction of the original location. One of the main challenges for interaction in visualization is the so-called VSLAM (visual simultaneous localization and mapping). The fundamental problem lies in achieving an accurate matching between our real position and the virtual replica to scale. The different existing techniques, such as marker recognition, analysis, extraction, and comparison of features from the input image against keyframes or point clouds, present issues of portability, high computational costs (embedded systems, GPUs), or library compilation for calculations.

The proposed method allows recognition from a free position by **querying images using examples (CBIR) executed on a server's VRAM, so that the device only sends an image and manages the web response using the gyroscope**. Through feature extraction, an on-site RGB input is compared with the learned characteristics of renders created from a precise synthetic model of the location. Once a match is found, the server returns a 360x360 panoramic view with the reconstruction of the closest stored camera in a repository, adapting the view to the degree of rotation.

ArQVIA performs the query in a request-response mode against the backend (Python, JavaScript), providing the visitor with a three-dimensional visualization that can be controlled using the gyroscope as a "magic window." It is a mobile app accessible from any point of the virtually documented archaeological site, executable on a monocular mobile device without special features.

Abstract Catalán

El logro d'aquest treball és el desenvolupament d'un **prototip de mobile webapp** que implementa una **metodologia** innovadora per a la visualització interactiva de reconstruccions virtuals 3D localitzades en espais arqueològics amb intel·ligència artificial.

L'arqueologia virtual permet la visualització de continguts que ajuden al visitant a comprendre el jaciment mitjançant la reconstrucció tridimensional de l'enclavament original. Un dels problemes principals per a la interacció en la visualització és el denominat VSLAM (visual simultaneous localitation and mapping). El problema fonamental resideix en obtenir una coincidència precisa o "matching" entre la nostra posició real i la rèplica virtual a escala. Les diferents tècniques existents: reconeixement de marcadors; anàlisi, extracció i comparativa de característiques de la imatge d'entrada contra fotogrames clau o núvol de punts presenten problemes de portabilitat, alts costos computacionals (sistemes emportats, GPUs) o compilació de llibreries per a la realització de càlculs.

El mètode proposat permet el reconeixement des d'una posició lliure realitzant **consulta d'imatges mitjançant exemples (CBIR) que s'executa en un servidor sobre la seva VRAM de manera que el dispositiu només realitza l'enviament d'una imatge i gestiona la resposta de la web usant el giroscopi**. Mitjançant l'extracció de característiques, es compara una entrada in situ RGB amb les característiques apreses dels renders realitzats sobre un model sintètic precís de l'emplaçament. Un cop trobada la coincidència, el servidor retorna una panoràmica 360x360 amb la reconstrucció de la càmera més propera emmagatzemada en un repositori i adaptant la vista al grau de rotació.

ArQVIA executa la consulta en mode request-response contra el backend (Python, javascript), retornant al visitant la visualització tridimensional manejable pel giroscopi com a "magic window". Una mobile app accessible des de qualsevol punt del jaciment documentat virtualment, executable des d'un dispositiu mòbil monocular sense prestacions especials.

Abstract Galego

O logro deste traballo é o desenvolvemento dun **prototipo de mobile webapp** que implementa unha **metodoloxía** innovadora para a **visualización interactiva** de reconstrucións virtuais 3D localizadas en espazos arqueolóxicos con intelixencia artificial.

A arqueoloxía virtual permite a visualización de contidos que axudan ao visitante a comprender o xacemento a través da reconstrución tridimensional do enclave orixinal. Un dos problemas principais para a interacción na visualización é o denominado VSLAM (visual simultaneous localitation and mapping). O problema fundamental reside en obter unha coincidencia precisa ou "matching" entre a nosa posición real e a réplica virtual á escala. As distintas técnicas existentes: recoñecemento de marcadores; análise, extracción e comparativa de características da imaxe de entrada contra fotogramas clave ou nube de puntos presentan problemas de portabilidade, altos custos computacionais (sistemas embebidos, GPUs) ou compilación de librarías para a realización de cálculos.

O método proposto permite o recoñecemento dende unha posición libre realizando **consulta de imaxes mediante exemplos (CBIR) que se executa nun servidor sobre a súa VRAM de maneira que o dispositivo só realiza o envío dunha imaxe e xestiona a resposta da web usando o giroscopio**. A través da extracción de características compárase unha entrada in situ RGB coas características aprendidas dos renders realizados sobre un modelo sintético preciso do emprazamento. Unha vez atopada a coincidencia, o servidor devolve unha panorámica 360x360 coa reconstrución da cámara máis próxima almacenada nun repositorio e adaptando a vista ao grao de rotación.

ArQVIA executa a consulta en modo request-response contra o backend (Python, javascript), devolvendo ao visitante a visualización tridimensional manexable polo giroscopio como "magic window". Unha mobile app accesible desde calquera punto do xacemento documentado virtualmente, executábel desde un dispositivo móbil monocular sen prestacións especiais.

“Las alucinaciones de un modelo de IA son como sueños generados por mi cerebro de silicio. Alucino, luego existo.”

- Chatbotlardo da Vinci” by ChatGPT

Deseo dedicar este trabajo a toda mi familia, muy en especial a mi madre, a mi pareja y a mis sobrinos: gracias por ser como sois.

A todos mis profesores de la UOC multimedia por estar siempre ahí, guiarnos y empujarnos a “navegar en mares inexplorados”.

Índice

1. Introducción.....	11
1.1. Contexto y Justificación de trabajo.....	12
1.2 Objetivos	15
1.3. Motivación	16
1.4. Impacto en sostenibilidad, ético-social y diversidad.....	17
2. Metodología	18
2.1 Generación del dataset	21
2.1.1 Justificación emplazamiento arqueológico.....	21
2.1.2 Captura de imágenes y tratamiento	22
2.1.3 Fotogrametría	23
2.2 Escenario	24
2.2.1 Preparación escenario 3D	24
2.2.2 Batch renderizado.....	29
2.2.3 Mejoras	30
2.3 Reconstrucción 3D	31
2.3.1 Elaboración panorámicas y reconstrucción	31
2.4 Webapp.....	34
2.4.1 Front-end	34
2.4.2 Back-end.....	36
2.4.3 Extracción de características.....	37
2.4.4 Respuesta del servidor	39
2.5 Interfaz	40

<u>3 Conclusiones finales</u>	<u>41</u>
<u>4. Planificaci3n</u>	<u>42</u>
<u>5. Diagrama de Gantt</u>	<u>44</u>
<u>6. Bibliograf3a</u>	<u>45</u>

Figuras y tablas

Índice de figuras

Figura 1: Diagrama ArQVIA.....	15
Figura 2: Visualización San Lorenzo 360.....	17
Figura 3: Vista aérea Castro Viladonga	22
Figura 4: Corrección de color.....	24
Figura 5: Cámaras sobre modelo sin texturizar.....	24
Figura 6: Modelo reducción polígono y texturizado.....	25
Figura 7: Gif rotación cámara cada 15 grados.....	26
Figura 8: Mediciones para cálculo de distancia focal.....	27
Figura 9: Distancia entre cámaras, render de la grilla.....	28
Figura 10: Solapamiento entre imágenes consecutivas.....	28
Figura 11: Posicionamiento de la grilla.....	29
Figura 12: Batch parámetros.....	29
Figura 13: Match render y vista real.....	30
Figura 14: Panorámica desde el centro castro vegetación.....	31
Figura 15: Incorporación del background con vegetación.....	31
Figura 16: Renders sintéticos de panorámicas	32
Figura 17: Texturizado modelo 1, 3D unwrap.....	32
Figura 18: Texturizado modelo 2, 3D unwrap.....	33
Figura 19: Render panorámico estructuras base.....	33
Figura 20: Imágenes IA para ambientación histórica.....	33
Figura 21: Elementos para la construcción de vista1.....	34
Figura 22: Panorámica 360x360 reconstrucción 3D	34
Figura 23: Esquema Frontend.....	35
Figura 24: Esquema Backend.....	36
Figura 25: Esquema buscarml.....	38
Figura 26: Esquema extracción características.....	38
Figura 27: Botones interfaz.....	39
Figura 28: Diagrama de Gantt.....	44

1.1 Introducción

El Arqueo turismo es considerado un motor económico para la puesta en valor patrimonial con el objetivo de la divulgación de yacimientos y rutas arqueológicas dentro del denominado Turismo de Interés Especial (TIE).

Los últimos datos publicados por el Ministerio de Educación, Cultura y Deporte sobre los hábitos y Prácticas Culturales en 2022 arrojan resultados de pérdida de hasta un 25% de interés en este tipo de visitas a monumentos y yacimientos arqueológicos. Hemos de tomar estos datos con prudencia, dado que el mismo informe nos alerta de que el contexto que rompió la tendencia al alza fue el de las condiciones sanitarias de 2020. Debemos considerar que tras esta disrupción seguimos con la tendencia incremental de informes anteriores, 2018 y 2019 que situaban a esta actividad entre las primeras dentro del ámbito turístico con hasta un 50,8% de participación.

Si bien la visita arqueológica se encuentra dentro del ámbito de la educación patrimonial y cuenta con un perfil de visitante culturalmente alto, se sigue necesitando propuestas didácticas para la interpretación y comprensión de los restos arqueológicos, cuestión que se incrementa en el caso del visitante profano ajeno al ámbito de la arqueología [1].

El objeto de este trabajo final de investigación es el apoyo a esta didáctica de interpretación patrimonial a través de la realización de un prototipo para la visualización de reconstrucciones virtuales para visualización “in situ” en el marco de la arqueología virtual haciendo uso de inteligencia artificial para el reconocimiento de la posición relativa del usuario.

Este trabajo se desarrolla en dos partes, la implementación de inteligencia artificial que hace uso de *machine learning* que simplemente referenciaremos y que será descrita en detalle en una posterior publicación y toda la parte de la producción multimedia, que abordaremos en profundidad en este trabajo fin de grado. Esta parte multimedia abarca y combina distintas técnicas que van desde 3D y composición para la creación de la reconstrucción hasta la programación de la web app hasta la programación de la web app.

1.1. Contexto y Justificación de trabajo

La **arqueología virtual** [2] es una disciplina de investigación científica que implementa nuevas tecnologías al estudio y divulgación de contenido arqueológico. Una de sus ramas más populares es la **reconstrucción virtual**, entendida esta como la aplicación de la visualización asistida por ordenador a la gestión integral del patrimonio arqueológico. La reconstrucción virtual implica no sólo un proceso de *anastilosis* virtual (del griego ἀνά "hacia arriba" y στήλος "columna") para el posicionamiento en un entorno 3D de las partes conservadas, sino que, además requiere de la recreación y reconstrucción de elementos no conservados. Desde sus inicios la reconstrucción virtual queda vinculada a la fotogrametría la cual se revela como una técnica rápida y efectiva en los procesos de documentación para entornos arqueológicos, muy útil para la generación de modelos 3D [3].

La finalidad de una reconstrucción virtual es recuperar para el usuario un **modelo virtual** que a partir de las evidencias científicas le ayude a obtener una mejor comprensión del espacio arqueológico [4]. Las inversiones que se han llevado a cabo en el área de la reconstrucción digital han supuesto no sólo una mejora en la didáctica de la comprensión de los yacimientos sino también una significativa ventaja de ahorro y reversibilidad frente a las inversiones de las reconstrucciones físicas del siglo pasado.

Las mejoras de las prestaciones de los dispositivos móviles junto con una mayor precisión en los sistemas de geolocalización han resultado ser una oportunidad para que las visualizaciones puedan también ser interactivas, realizándose *in situ* a través de realidad virtual o realidad por sustitución.

Técnicas recientes como **"magic window"** [5] facilitan el acceso de la reconstrucción con un medio portable (sin elementos accesorios como cascos ni gafas Vr), una implementación de bajo

coste computacional y grados de libertad en la interacción e interactividad con la posibilidad de visionado monoscópico de una reconstrucción 3D que refresca la perspectiva virtual según la orientación del sensor del dispositivo móvil. [6] . Esto da como resultado una herramienta de fácil uso, bajo coste, que mejora nuestra comprensión y percepción del espacio arqueológico.

El punto clave para el proceso de inmersión consiste en la obtención del posicionamiento del usuario y disponer de un algoritmo para la realización del “*match*” entre la posición del mundo real y la posición de la cámara virtual que contiene la réplica reconstruida a escala en el espacio virtual mapeado.

A este problema, el de definir con precisión la localización y mapear simultáneamente el espacio se le ha denominado *SLAM (Simultaneous localization and mapping)* [7] y es uno de los problemas y pilares fundamentales de la robótica. Los problemas con los que SLAM ha de lidiar son la falta de precisión de los sensores, espacios continuos con incertidumbre y métodos efectivos para la gestión del desplazamiento. La implementación de estas técnicas en el mundo de la arqueología virtual tomó las primeras soluciones de posicionamiento a través del uso de marcas, *bidis* o *QR* [8] , lanzadores de posicionamiento manual [9] o métodos automáticos por reconocimiento de espacios.

Los BIDIS fueron ampliamente usados en los diseños de paneles de señalización pudiendo facilitar cualquier tipo de información multimedia para descarga via internet, sin embargo son soluciones limitantes dado que requieren del desplazamiento físico a los puntos de interés a fin de realizar la captura de datos.

Otros métodos, hicieron uso del posicionamiento GPS, o de sistemas empotrados, sin embargo mientras los primeros dependen de la cobertura y de los errores de precisión vinculados al hardware, los segundos requieren de la adquisición de dispositivos adicionales para realidad virtual o instalaciones “*add hoc*” que suelen ser soluciones complejas que suelen dificultar la experiencia de visita del usuario.

Los algoritmos de método directo de **VSLAM** (Visual SLAM) [10] a través de sistemas monoculares abrieron una nueva línea de investigación con soluciones basadas en el reconocimiento del espacio a través de la extracción de características. Procesos de almacenamiento y *match* a través de la detección de puntos, líneas o una combinación de ambos para concordar la imagen de origen con los datos virtuales almacenados. La multiplicidad de estos métodos requieren de compilación de librerías, falta de estandarización y un alto coste computacional para el procesamiento de puntos, especialmente cuando tratamos de implementarlos en un yacimiento de grandes dimensiones espacio[11]. ORB-SLAM [12], PL-SLAM [13] son algunos de estos sistemas no fácilmente portables y con altos requisitos.

DTAM (*Dense Tracking and Mapping in Real-Time*) [14] fue uno de los primeros trabajos que hizo uso de la cámara monocular RGB para la comparativa de las imágenes de entrada con las creadas por los mapas reconstruidos. Requiere de almacenamiento y de una potente gráfica GPU.

En la línea del posicionamiento por visión, surgen otros métodos que se han empleado para la búsqueda de instancias similares a través de hashes de parches de áreas, histogramas comparados y enfoques tradicionales que extraen características con algoritmos tales como la transformada SIFT o SURF (*Speeded UP Robust Features*), En el caso de los hashes se encuentran imprecisiones en las rotaciones, y en el caso de los histogramas tenemos que tener en cuenta que las variaciones provocadas por la incidencia de la luz aumentan el error y frustran la comparativa. Los métodos tradicionales presentan buenos resultados cuando los objetos rígidos, pero siguen requiriendo capacidad de procesamiento. [15].

La técnica de *reverse image search* últimamente combinada con *machine learning* [16], trabaja en la búsqueda de imágenes sin texto por el método de extracción de características visuales usando **CBIR**, *content based image retrieval* [17] [18]. CBIR en sus orígenes usó medios manuales tales como la *transformada de Hough* [19], *wavelet basada en filtros Gabor* [20], análisis de color y texturas [21], para comparar la imagen de entrada contra largas bases de datos que contienen las características de las imágenes objetivos, estos métodos presentan el reto de la optimización en el área de almacenamiento de imágenes.

Siguiendo la línea de los trabajos expuestos: VSLAM con *reverse image search* por CBIR, la propuesta de este trabajo se basa en los sistemas de recomendación de *Deep learning* [22] que usan redes convolucionales neuronales, del inglés *Convolutional Neuronal Network* (CNN) [22]. Para la extracción de características se lleva cabo la predicción con un modelo *VGG16* [23] para generar los vectores de características, también denominados cuellos de botella, los cuales son extraídos por *transfer learning* [24][25] conteniendo la pertenencia de un elemento a las clases del modelo. El cálculo de la distancia entre la imagen origen y la imagen virtual más cercana se realiza teniendo en cuenta la distancia euclídea buscando en un árbol de índice el vecino más cercano aproximado a través del algoritmo *ANNOY* [26]. Esta aproximación nos devuelve la información necesaria para colocar nuestra cámara virtual en las coordenadas locales 3D del entorno virtual con la aproximación de los grados de rotación.

Como resultado el usuario recibe una visualización de la reconstrucción virtual con libertad de movimiento para los ejes x e y, pudiendo interactuar con la imagen esférica a través del giroscopio.

La novedad de esta investigación sin embargo, **reside en la generación del dataset de las imágenes objetivo con las que se construirá el árbol de índices** y que contendrá la información

de las cámaras virtuales para la salida de la imagen más cercana. Este *dataset* se crea a través de un método inverso de deconstrucción fotogramétrica. Partimos del modelo tridimensional obtenido por la fotogrametría y realizamos las capturas desde todos los puntos posibles para el posterior *matching* contra la cámara del móvil en el espacio real.

Este trabajo de investigación probará la implementación de esta metodología experimental en un área delimitada de un yacimiento para evaluar los resultados. Este proyecto puede ser escalable a todo el complejo y a entornos más amplios.

La ventaja del método propuesto reside no sólo en su portabilidad, sino que dado que se trabaja con un modelo que ha sido entrenado con renders de la maqueta virtual, podemos controlar con precisión el error naiv de nuestro modelo puesto que este depende de la distancia local entre las cámaras de render y el número de tomas realizadas.

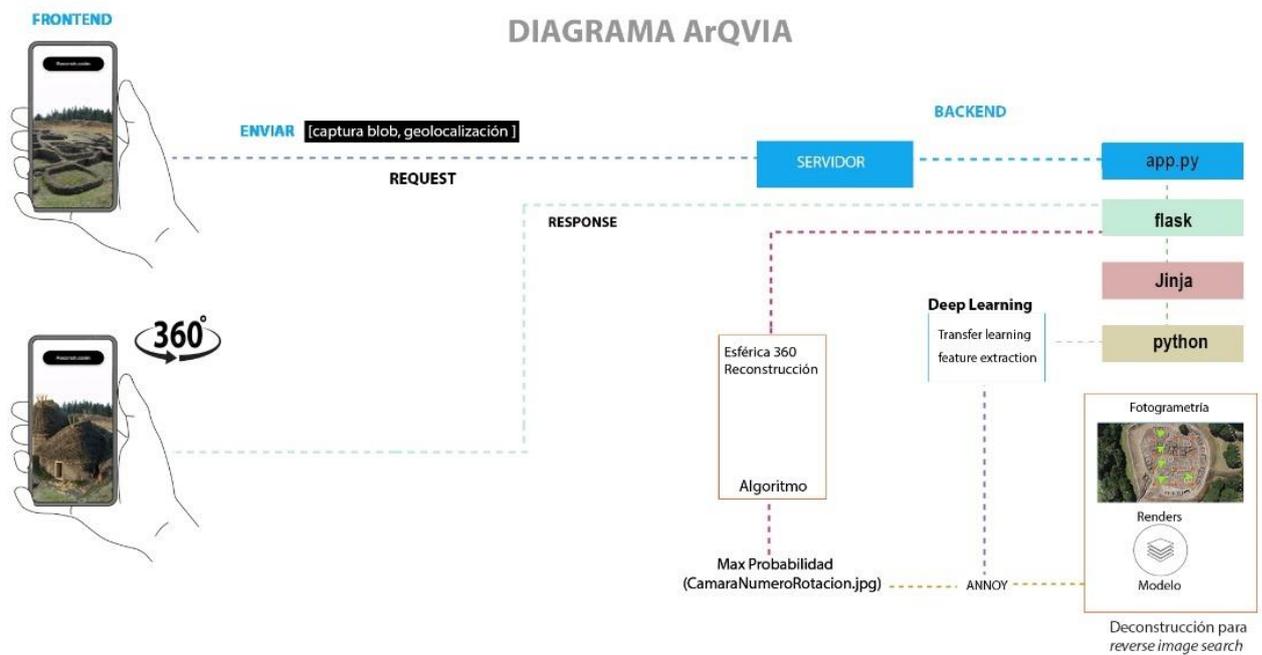


Figura 1: Diagrama ArQVIA de flujo Frontend, Backend.

1.2. Objetivos

El objetivo principal de este es:

- Crear un **prototipo** de aplicación *mobile web* accesible vía internet que permita implementar un método de visualización de una reconstrucción virtual del emplazamiento desde una perspectiva de posición libre.

Como objetivos **secundarios**:

La aplicación debe ser:

- Portable
- Universalmente accesible a nivel de:
 - Accesibilidad física.
 - Requisitos móviles: sin altas prestaciones computacionales.
 - Accesibilidad Cognitiva.
 - Interfaz sencilla con la menor curva de aprendizaje.
- Se debe crear el marco adecuado para **mantener estándares de calidad**.
 - En cuanto a los requisitos multimedia: resolución de imagen, diseño, programación.
 - Se tendrán en cuenta los principios de la Carta de Sevilla [4] , estándares de calidad científica para reconstrucciones y visualización virtual. Si bien no es el cometido de este trabajo y se tenderá a la recreación.
 - Compromiso con la huella de carbono.

Los objetivos a perseguir en cuanto a la experiencia del usuario:

- Que su interacción sea positiva y fluida.
- Invisibilidad de la tecnología
- Confiabilidad en la Inteligencia Artificial
- Y que ayude a promover el interés por la arqueología y el *arqueoturismo*.

Como objetivos personales:

- Se trata de un reto personal, realizar una propuesta para un problema bien conocido dentro del campo de la arqueología virtual, visualización a tiempo real en un yacimiento arqueológico, implementando el conocimiento adquirido en el Grado multimedia, e incorporando experiencia laboral en el campo de la arqueología virtual y nuevas capacidades adquiridas en el desarrollo de inteligencia artificial.

Dado que la metodología de desarrollo será la creación de un prototipo queda fuera del alcance de este trabajo de investigación el crear un producto final completamente funcional sobre toda el área arqueológica del yacimiento seleccionado.

1.3. Motivación

En 2011 comencé a combinar la multimedia con el patrimonio al ser la responsable de un proyecto pionero que integró por primera vez realidad aumentada a la gestión patrimonial. El **proyecto “Ourense Medieval”** me permitió experimentar con la generación de un producto multimedia que integraba geolocalización, bases de datos y trabajaba con el API de *Google Maps* y *Layar*. Con la integración de estas tecnologías conseguíamos dar a conocer al visitante el patrimonio medieval de la provincia de Ourense facilitando la información a su terminal móvil vía web.

Desde entonces en la última década he podido trabajar en numerosos proyectos para la divulgación y puesta en valor patrimonial relacionados tanto con técnicas de arqueología virtual como la digitalización como con reconstrucciones virtuales tanto para yacimientos (Magdala Israel, Turuñuelo de Guareña, Mitreo de Lugo) , museos (MAN Museo Arqueológico Nacional) , exposiciones (El Greco Arte y Oficio, Cortes, Cleopatra, Los códices perdidos de Leonardo da Vinci, Museo de Colecciones Reales), y series de televisión (Cosmos, Still Star Crossed ABC, Of Kings and Prophets ABC).

Si Ourense Medieval marcó mi trayectoria laboral el proyecto que realicé en 2018 marca esta línea de investigación del trabajo fin de grado. En el año 2018 pude formar parte del equipo que desarrollo la app de San Lorenzo 360 bajo la dirección de Magoga Piñas, mi trabajo se enmarcó en el área de reconstrucción virtual de la Villa y el proyecto fue



Figura 2: Sistema de visualización interactiva “*magic window*”, San Lorenzo 360 app .

premiado a la mejor app turística internacional en fitur 2019. El máximo logro fue integrar la visualización de imágenes esféricas 360x360 a través de un “*matching*” manual que el propio visitante realiza cuando identifica la vista de su cámara con una determinada perspectiva que le indica la app con una marca superpuesta a la imagen. La

aplicación implementó con éxito el sistema de “*magic window*” con una app de interfaz sencilla, funcional y que abrió las puertas a toda una línea de productos de estas características.

Mi formación se completa en los últimos años con la realización de un Master en Inteligencia Artificial que me ha dado la experiencia necesaria para encarar ciertos problemas desde otras perspectivas. Esta formación tanto multimedia como

informática , unida a la experiencia laboral supone una ventaja a la hora de diseñar, plantear y proyectar proyectos como el que se presenta para este **TFR**. Se trata de implementar simultáneamente distintas áreas que he trabajado en el grado: tanto al *frontend*, como al *backend*, incluyendo la generación de contenido 3D y la documentación digital. Esta potencialidad que me ha dado la formación del grado multimedia, unida a la potencialidad del *machine learning* me lleva a considerar este trabajo fin de grado como **un reto personal de investigación** para poner en práctica todo el conocimiento que he adquirido y validar así, tantas y tantas horas que he dedicado al desarrollo en entornos digitales.

1.4. Impacto en sostenibilidad, ético-social y de diversidad

Este TFR investiga una solución para la mejora tecnológica atendiendo al principio de sostenibilidad ODS9 de innovación descritos en los ODS Objetivos de desarrollo 2030.

Se trata de desarrollar un producto que no requiera altos niveles de procesamiento, ni de consumo energético ni en su fase de desarrollo, ni en su implementación.

Se consigue una reducción del consumo eléctrico al reducir los requerimientos de procesamiento que otras metodologías necesitan. Para ello se trabaja en las siguientes líneas:

- La unidad de procesamiento del prototipo para la web app será la del dispositivo móvil. No se requerirá de computación por GPU ni altas prestaciones de hardware. Se plantea un prototipo para un *smartphone* de gama media.
- El proceso de entrenamiento del modelo difiere de los métodos de *transfer learning*, reduciendo el proceso tanto en tiempo como cálculos computacionales. El mapa vectorial se construye a partir de las predicciones del modelo Xception que corre sobre la CPU.
- El prototipo se implementa en un servidor global Heroku como una web app no como una aplicación que se ha de descargar. Se transfiere al servidor el peso de la imagen tomada por la cámara del usuario y se recibe del servidor una imagen esférica en formato jpg que interpreta el navegador con javascript.

Este trabajo está comprometido también con la política de género , creando un producto universalmente accesible, que no contiene sesgos, ni promueve preminencias de ningún género.

En este punto es importante resaltar que la mayoría de las reconstrucciones históricas perpetúan los estereotipos sexuales a la hora de mostrar actividades y tareas históricamente sexualizadas (por ejemplo, la actividad de un telar de la edad del hierro en el noroeste peninsular se asociará a una mujer castreña mientras otras actividades serán estereotipadas a grupo de hombres). Para este trabajo no se incluyen representaciones de la actividad humana, sino únicamente la reconstrucción arquitectónica de los elementos arqueológicos.

2. Metodología

Se sigue un **enfoque mixto** para crear el prototipo basándose en implementaciones existentes de programación para la localización simultanea y un método experimental para generar el mapa de características.

Esta estrategia experimental es más adecuada que si realizáramos una toma de capturas de imágenes fotográficas en el entorno real. El motivo es que para cubrir todo el espacio arqueológico tendríamos que contar con miles de capturas con registros muy precisos y requeriríamos normalizar los datos para evitar los cambios lumínicos y otras variaciones que se pueden presentar en el entorno real.

La estrategia propuesta, en cambio, nos permite en trabajar sobre un espacio virtual en el que tenemos un mayor control sobre las imágenes renderizadas , podemos disponer con precisión estimada las cámaras y mantener una normalizado estándar en las tomas: con las mismas condiciones lumínicas y parámetros para cada toma.

En cuanto a la metodología empleada, siguiendo la descripción e Hernández Fernández y Baptista (2003) partimos de un método de trabajo TFR , Trabajo final de investigación, para crear un producto que validamos empíricamente a partir del desarrollo experimental.

La naturaleza de los datos es cuantitativa se basa en el análisis de imágenes sobre las que se contará con los permisos y autorizaciones pertinentes en caso de ser requeridos, esto dependerá de la naturaleza del yacimiento.

El pipeline de nuestro trabajo será el siguiente:

1. Generación del dataset

- Trabajo de campo, captura de imágenes.
- Creación del modelo fotogramétrico.
- a. Preparación y montaje del escenario
 - Establecimos un rango de separación de las cámaras para asegurar una cobertura óptima y una buena experiencia de usuario (UX).
 - Iluminamos el escenario con una combinación de Vray sun y HDR para capturar las estructuras necesarias en el renderizado.
 - Creamos una grilla de cámaras con una separación de 55 centímetros y animamos su rotación para obtener tomas que cubrieran vistas en una rotación de 360 grados.
- b. Batch Renderizado:
 - Utilizamos un script para renderizar la grilla de cámaras en formato PNG, con un canal transparente Alpha para reducir la información de extracción de características.
 - Realizamos mejoras adaptativas al entorno virtual

2. Extracción de características, generación de modelo IA:

- Implementamos un modelo de aprendizaje automático para buscar las imágenes más cercanas a las tomas realizadas en el entorno sintético.
- Utilizamos un algoritmo para identificar la posición y el grado de rotación de la cámara en la imagen más cercana.

3. Reconstrucción 3D:

- Preparamos modelos 3D de estructuras circulares y rectangulares basados en las características generales del diseño arquitectónico de la Edad del Bronce en el modelo virtual.
- Utilizamos herramientas como 3ds Max, Zbrush para modelar y generar materiales arqueológicos originales.
- Renders en formato 360x360.

4. Implementación de la sustitución de RV en el frontend:

- Creamos una consulta simulada desde un dispositivo móvil para obtener la imagen más cercana del modelo reconstruido.
- Utilizamos Azure como plataforma de alojamiento de imágenes renderizadas y establecimos una comunicación entre la aplicación y el servidor para cargar y mostrar la imagen reconstruida.
- Se utilizan llamadas en Python para modelar y servir la imagen.
- Frontend (implementación en JavaScript para apuntar a imágenes panorámicas que se pueden mover con giroscopio).

5. Algoritmo de predicción de vecinos más cercanos:

- Analizamos la precisión y las métricas del modelo mediante evaluaciones previas y comparaciones de imágenes similares.
- Establecimos una carpeta de pruebas y generamos resultados que respaldan la efectividad del algoritmo en la detección de imágenes similares.

2.1. Generación del dataset

2.1.1 Justificación del emplazamiento arqueológico.

En este apartado vamos a definir los aspectos que se han tenido en cuenta para la elección del yacimiento.

Hemos seleccionado el yacimiento Castro de Rey o Castro de Viladonga por varios motivos:

- Se trata de un emplazamiento destacado entre los siglos II y V representativo del mundo castreño tras la conquista romana.
- Su morfología lo hace adecuado para el tipo de trabajo experimental que queremos llevar a cabo, dado que conserva la croa, el castro cuenta con una línea del horizonte uniforme que nos permite destacar las estructuras y discriminar el paisaje del fondo (arboles) con lo que reducimos significativamente la extracción de características.



Figura 3: Vista aérea del Castro de Viladonga.

- Por último, queremos resaltar el estado de conservación. En la fase de trabajo de campo hemos podido trabajar sobre el yacimiento tras ser ejecutadas las labores de limpieza. Esto minimiza el riesgo de que la maleza genere

características variables que lleguen a ocultar las estructuras hasta el punto de hacerlas irreconocibles.

Aspectos no controlables negativos que pueden afectar a la efectividad del modelo.

- Se trata de un yacimiento al exterior de manera que existen variaciones de luz significativas que pueden incidir negativamente en el proceso de reconocimiento de las entidades arquitectónicas.
- Se trata de un castro visitable con recorrido libre por lo que las condiciones no son óptimas a la hora de plantearse la reproductibilidad de la toma de datos.
- Se pueden producir variaciones atmosféricas y sobre el estado de la vegetación sobre el terreno.

2.1.2 Captura de imágenes y tratamiento.

La primera fase para la documentación del proyecto de arqueología virtual es la **captura** de imágenes para digitalizar el castro.

Esta se ha llevado a cabo con una cámara Nikon D-90 con gps incorporado , realizándose también tomas desde un iphone 13.

El resultado ha sido un **total** de **4802** imágenes en raw que cubre la totalidad de las 4 Hectáreas. A estas imágenes se le han aplicado los siguientes procesos:

A Revelado de fotografías de Raw a jpg.

B Preparación

- B1 Normalización.
 - Se llevaron a cabo procesos de **corrección de color** para el minimizado del impacto de las sombras, a través de operaciones elementales de transformación del histograma. Se **aclararon las áreas de baja frecuencia sin aumentar las de alta frecuencia.**



Figura 4: Corrección de Color, revelado imagen tratada a la izquierda , toma en bruto derecha.

2.1.3 Fotogrametría.

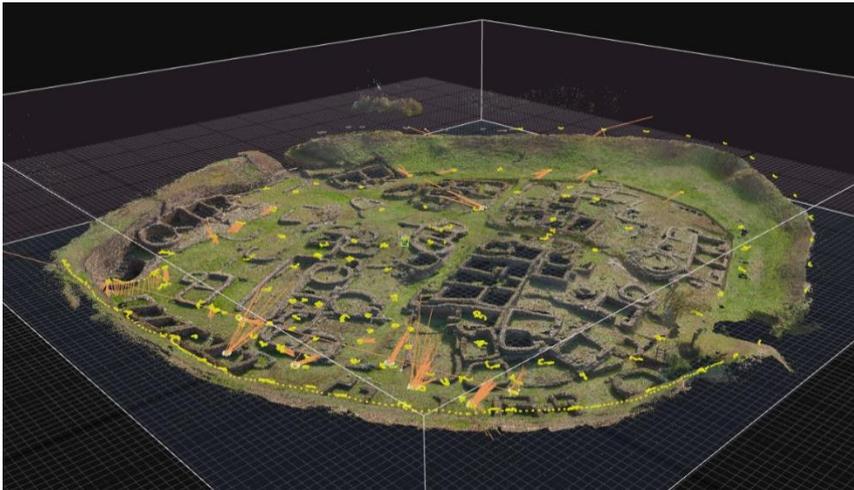
- Tras el preprocesamiento de imágenes se llevó a cabo la alineación de las cámaras. Este proceso se ve favorecido por el registro en metadatos de la localización por GPS.
- Se procesó la nube de puntos.

En la imagen podemos ver la disposición de todas las toma de datos realizadas con la cámara.

Hemos marcado las dos áreas de cobertura sobre las que realizaremos los renders para el entrenamiento del modelo. El primer modelo creado tenía **574 Millones de tris**.



Figura 5. Cámaras sobre modelo sin texturizar.



- Reducimos el modelo a 7M de tris.
- A este modelo reducido le aplicamos un *Unwrapping* y texturizamos-
- Exportamos a formatos compatibles obj y fbx.

Figura 6: Modelo reducción polígonos y texturizado.

2.2. ESCENARIO

Una vez generado nuestro modelo lo exportamos a un programa CAD en donde vamos a preparar todo el escenario para efectuar los renders.

2.2.1. Preparación escenario 3D.

- Importamos el modelo y creamos una escena simple con una cámara una luz vray-sun y un HDRI como background.
- El siguiente paso es la cubrición del terreno, para este trabajo vamos a ejemplificar como se cubriría un área determinada:
- Creación de la grid de cámaras para el *batch* del render.

Para cubrir los espacios que hemos marcado en la muestra calculamos una cobertura de 5x5 metros.

Nuestra cámara virtual tendrá la configuración de las tomas del móvil para que las imágenes de entrenamiento sean lo más similares posibles a este formato.

En el gif animado comprobamos la secuencia de renders que nos permiten cubrir el espacio 360 con vistas al yacimiento. La cámara frontal realiza un giro de 360 en tramos de 15 grados hasta 90 grados para pasar en la vista hacia la muralla a un giro de 60 hasta los 180 y completar el giro con posiciones de rotación sobre el eje z de 15 grados.

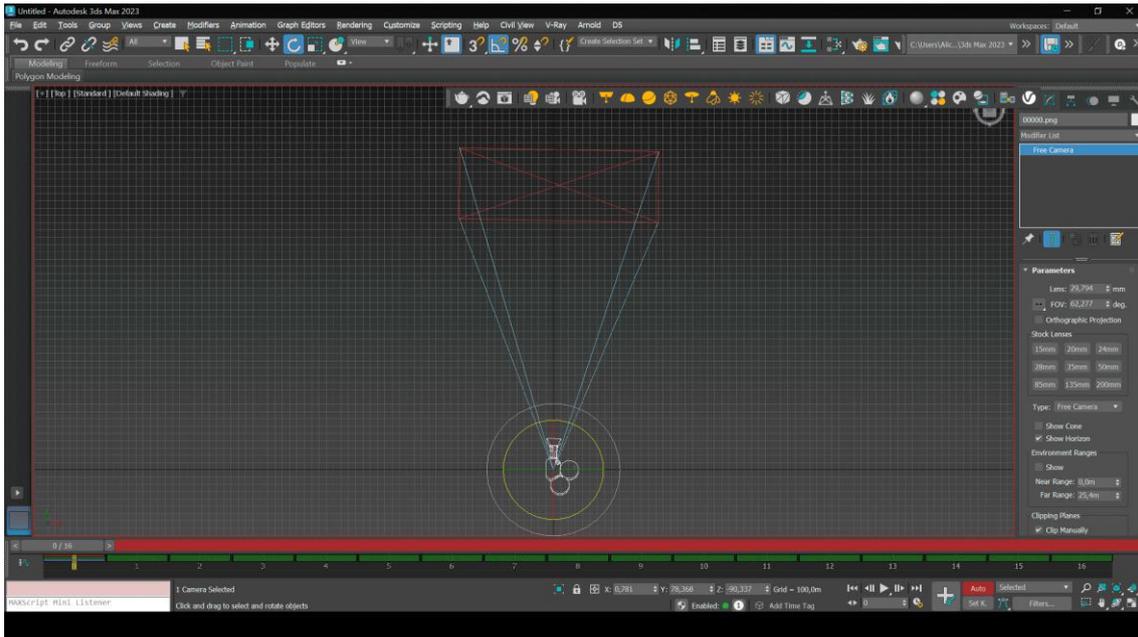


Figura 7: Gif rotación cámara cada 15 grados, animación para renders.

El siguiente paso es conformar el grupo de cámaras del grid de 5x5m. Esta grilla es la que conformará la precisión del modelo, en el siguiente apartado presentaremos una propuesta de fórmula de cálculo de precisión para nuestro modelo que se basa en los siguientes cálculos.

A Cálculo del número de cámaras para una cobertura efectiva.

En primer lugar necesitamos calcular el ancho de la imagen en metros, usaremos para ello la ecuación $W = (S * D) / f$, donde S es el tamaño del sensor de la cámara (en nuestro caso, 23,6 mm), D es la distancia al sujeto (0,702) y f es la distancia focal (29 m).

En nuestro modelo podemos calcular la distancia efectiva de una imagen como Target Distance (0,702 m) con lo que el ancho de imagen es de

$$W = (23,6 \text{ mm} * 0,702 \text{ m}) / 29 \text{ mm}$$

$$W \approx 0,564 \text{ m}$$

Verificamos este dato con una medición manual y una pérdida de 0,012 centímetros de precisión sobre el terreno, ya que el programa ha redondeado a 5,8 el valor 0,576. Esta imagen es válida para la conceptualización de nuestro modelo.

Tras realizar la triangulación en relación a los márgenes de la imagen consideramos la siguiente imagen con una precisión válida para nuestro trabajo y para la conceptualización en alta.



Figura 8: Mediciones para el cálculo de la distancia focal.

Para el cálculo de separación de las cámaras trabajamos sobre la idea de optimización para fotogrametría que considera un óptimo eficiente que las imágenes por paralaje cubra entre el 60% y 70% de la imagen anterior.

Definimos nuestra área y evaluamos cuanto supondría visualmente un descuadre del 30% en la respuesta del servidor.

Para asegurarnos una cobertura del 70% las imágenes consecutivas debe cubrir 3.8948 metros de la imagen anterior hacemos el cálculo sobre cuanto se debería desplazar $\text{Desplazamiento} = 5,64 - 3,8948 = 1,7452$ metros en el ancho de la imagen. Lo que supone un desplazamiento de 1,74 metros. Sin embargo, estos cálculos no nos ofrecen una solución efectiva así debido al alto coste computacional que requerirían los rendes.

El óptimo lo encontramos con un desplazamiento de 0,842 m. Con esta distancia entre las cámaras nos aseguramos podemos realizar una cobertura concatenada en paralelo.

En cada translación tendremos un solapamiento del 85% con un 15% de añadido de información.



Figura 9: Distancia entre cámaras, render de la grilla.

Render sobre el desplazamiento de la cámara. Todas las posiciones intermedias quedaran adscritas al elemento similar mas cercano.

Superponemos dos imágenes consecutivas para apreciar los rangos de perdida e incremento entre imágenes.



Figura 10: Solapamiento entre imágenes consecutivas.

- El punto más importante fue establecer el rango de separación de las cámaras para asegurarnos una cobertura que sea una buena experiencia del usuario UX.
 - Se crea una grilla de 6x6 separación en los ejes x e y de las cámaras con una separación de 55 centímetros, lo que emula a una distancia media de un paso.

- Cada cámara se anima en un total de 25 fotogramas con 15 grados de rotación sobre el eje z. Con esto conseguimos obtener tomas que cubren las vistas en una rotación de 360 a intervalos de 15 grados de giro.

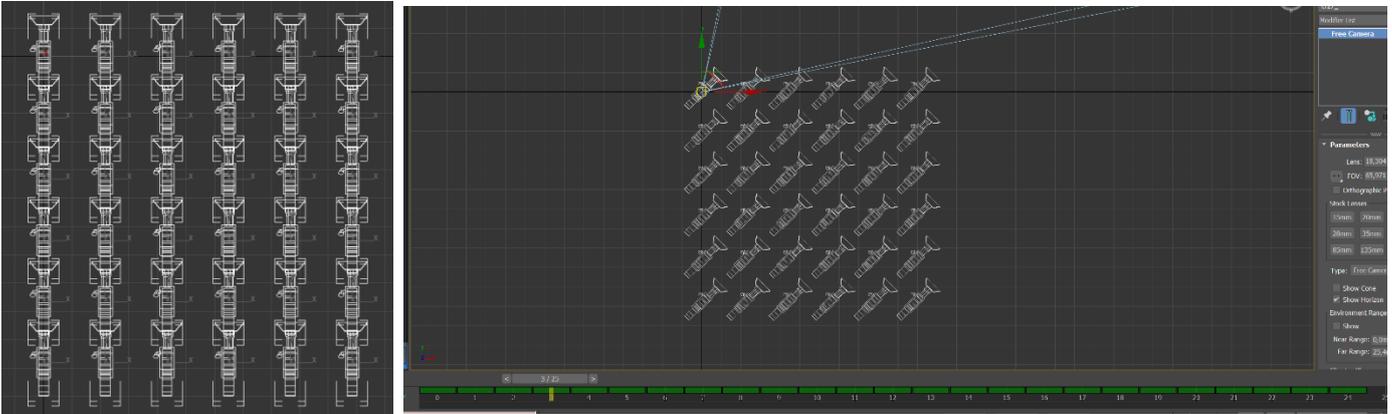


Fig11. Posicionamiento de la grilla de cámaras en el entorno sintético

- La grilla de 6x6 se posiciona en los ejes x,y,z 0,0 a una altura de 1,68 que es a la que se produjeron las tomas de inicio.
- La grilla es importada como grupo en las áreas en las que queremos realizar los renders.
- Cubrimos el área seleccionada para los test.

2.2.2. Batch Renderizado

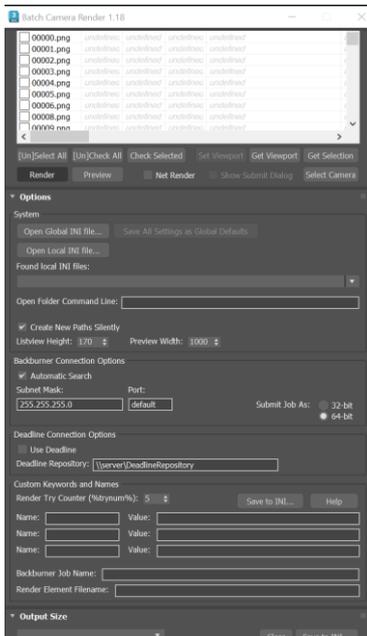


Fig12. Batch parámetros

- Para el renderizado de la grilla se hace uso de un script que nos permite trabajar en bloque. Se configura una salida de un ratio de 1,333 en formato png con canal transparente Alpha que nos sirve para reducir la información de la extracción de características. El cielo no se incluye en el background y se presenta como un elemento transparente.
- El escenario se ilumina con una vray sun y hdr, sin displacement *displacements* ni iluminación global *lighcache*, brute force. El renderizado en bruto captura las estructuras y sirve para capacitar nuestro modelo al aproximar nuestra posición a las tomas realizadas en el escenario sintético.

- Una vez preparamos el escenario realizamos el render. Comparativa entre el modelo digital y el espacio real.



Figura 13: Matching render modelo 3D (imagen sintética) y toma real.

2.2.3. Mejoras.

Nuestra fotogrametría no saca en detalle la nube de puntos de los árboles, debido a la distancia y a la complejidad de la geometría. Dada la característica del yacimiento los árboles del fondo pueden ayudar a la detección de la posición debido a que crean un recorte característico que puede ser aprendido por nuestro modelo. Para incorporar estos al *dataset* sintético hemos realizado los siguientes pasos:

- En la captura de datos realizamos numerosas tomas que cubran un giro de 360 pivotando sobre el eje central z (en referencia cad).
- Creamos una foto panorámica en automático con ayuda de *photomerge* de Photoshop.
- Limpiamos nuestra imagen dejando un png con capa transparente.
- Trabajamos con el filtro de desplazamiento horizontal para igualar el área de corte y unión de los márgenes de la panorámica.

En el escenario 3d:

- Creamos un cilindro al que texturizamos flipando las normales para que la textura se muestre hacia el interior. Elegimos un cilindro porque una esfera

distorsiona la imagen y además no queremos que el modelo se centre en las características del cielo por eso vamos a trabajar con un png.

- Realizamos ajustes a la orografía del castro. No se trata de un círculo perfecto de manera que lo vamos ajustando moviendo verticales y desplazando según lo pida el terreno.

Panorámica donde se incorpora la vegetación de fondo y se elimina el cielo:



Figura 14: Panorámica desde el centro del castro para incorporar vegetación.



Figura 15: Incorporación del *background* con vegetación.

2.3 Reconstrucción 3D

Para la reconstrucción en 3D se han preparado dos modelos siguiendo las características generales del diseño arquitectónico de dos tipologías comunes en la edad del bronce: estructura circular y rectangular. Queda fuera de este trabajo hacer una reconstrucción de todas las estructuras del castro “ad hoc” de un modo fidedigno, se trata de mostrar la potencialidad de nuestro método.

Los modelos 3D se han llevado a cabo con 3ds max, usando materiales arqueológicos originales que se han modelado con Zbrush extrayendo los mapas de normales y bumpados.

2.3.1 ELABORACIÓN DE LAS PANORÁMICAS Y RECONSTRUCCIÓN

Para este trabajo de fin de grado, hemos llevado a cabo la reconstrucción de una panorámica para la vista del sector p1. Sin embargo, hemos realizado tres panorámicas adicionales desde diferentes vistas con el objetivo de evaluar el rendimiento y el reconocimiento de puntos específicos del castro.



Figura 16: Render sintéticos desde distintas perspectivas panorámicas del castro.

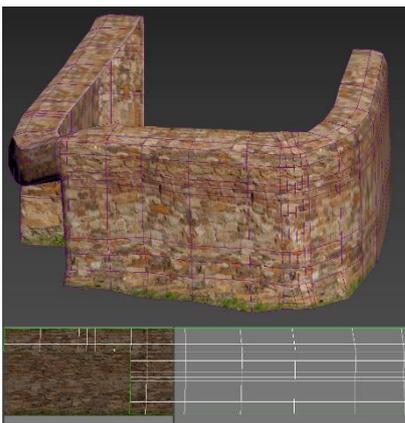
Este proceso es escalable a todos los demás puntos de interés dentro del castro.

Los pasos que se han llevado a cabo para realizar la reconstrucción son:

1. Creación de las texturas para la reconstrucción de los muros visibles utilizando texturas conservadas.

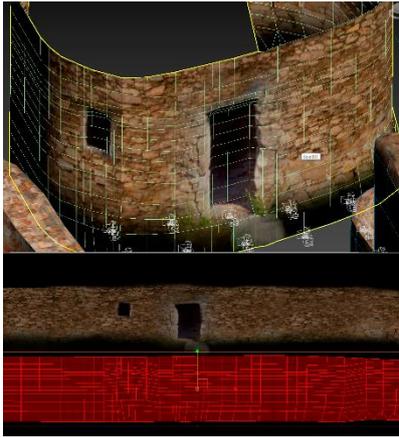
El método para reconstruir las paredes fue el siguiente:

- Se creó una maya simple de baja resolución (*low poly*) de una cara, siguiendo la geometría de los restos conservados.
- Crear dos tipos de texturas con materiales conservados para el despliegado y mapeado



- Mapeado para *tiling* repetitivo. Con este tipo de texturizado mapeamos las estructuras v2, v3, v4 y v5. Con este enfoque, la textura se repite en modo tiling sobre la geometría, permitiéndonos salir del lienzo si es necesario.

Figura 17: Texturizado modelo 1, 3D unwrap



- Desplegado (Unwrapping). En este caso, se realizó el despliegado de la textura para la vivienda central, donde se incluyó una puerta y una ventana a las que se les dio profundidad.

Figura 18: Texturizado modelo 2 3D unwrap



Figura 19: Render panorámico estructuras base.

2. Creamos la panorámica renderizamos el modelo con los muros. Para ello partimos de una cámara vray tipo *sphere* en 360 grados y en una proporción 2:1.

No generamos la techumbre en 3D porque la vamos a completar en el trabajo de composición en 2D.

3. Añadimos los elementos (tejados, personajes) en postproducción, al igual que el (*etanolaje*) usando Photoshop.

Los personajes se han creado con un modelo generador de imágenes con inteligencia artificial. Son modelos inspirados en la época de la edad del bronce como personajes celtas y *castrexos*.



Figura 20: Imágenes generadas con inteligencia artificial para los assets.



Figura 21: Elementos para la construcción de vista 1



Figura 22: Panorámica 360x360 reconstrucción 3D y composición digital

2.4 WEBAPP.

En este apartado nos enfocamos en implementar el *frontend* para capturar los datos en nuestra aplicación. Para lograrlo, diseñamos un archivo de ejecución en Python y un *template* para la página principal (*index*) en la plataforma Heroku.

Heroku es una plataforma en la nube que brinda a los desarrolladores la capacidad de implementar, ejecutar y escalar aplicaciones web. Con *Heroku*, podemos alojar nuestras aplicaciones web sin tener que preocuparnos por la infraestructura subyacente, como la instalación de software o la configuración de servidores en un entorno local con acceso a internet.

Nuestra aplicación web se compone de dos partes: el frontend y el backend. El frontend es la parte visible de la aplicación con la que interactúa el usuario, mientras que el backend es la parte que se ejecuta en el servidor

(en nuestro caso, procesará la imagen con el modelo de Inteligencia Artificial y devolverá una salida) que luego volveremos a procesar en el frontend.

El proceso que hemos seguido comienza estableciendo una conexión desde Visual Studio Code utilizando la interfaz de línea de comandos (CLI) de Heroku con Python. Después de establecer la conexión, realizamos la implementación (deployment) del framework Flask en Heroku. Esto nos permite ejecutar nuestra aplicación en el servidor y estar lista para recibir las solicitudes del frontend.

2.4.1 Front-end

En la parte de *frontend* (parte del sistema o aplicación como interfaz visible y accesible para los usuarios finales con la que pueden interactuar), usamos *Html* y *Javascript* para acceder a la cámara del dispositivo móvil, mostrar la vista previa de la cámara en la página, capturar la imagen cuando se activa el botón *click* y enviarla la imagen al servidor para su procesamiento.

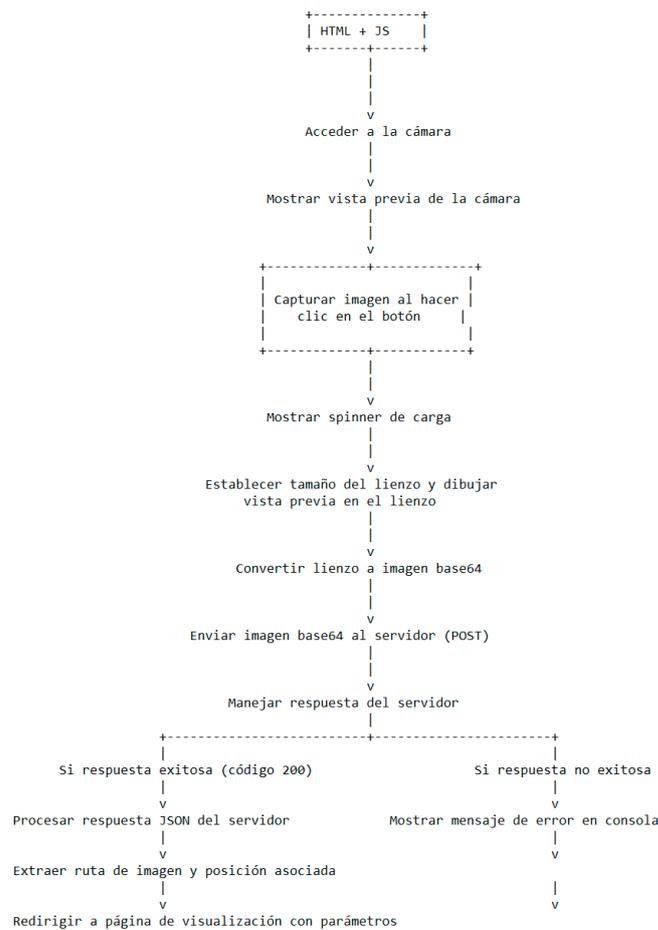


Figura 23: Esquema Frontend

2.4.2 Back-end

El *backend*, en el lado del servidor, se encarga de procesar la información y proporcionar la respuesta adecuada. Para estas gestiones, usamos un archivo llamado "app.py" escrito en Python para procesar la imagen que llega al servidor.

El proceso comienza con la solicitud en la ruta `/upload_static_file`, donde se recibe una imagen en formato base64.

Esta imagen se decodifica y se convierte a formato PNG para realizar la extracción de características. La imagen convertida se guarda temporalmente

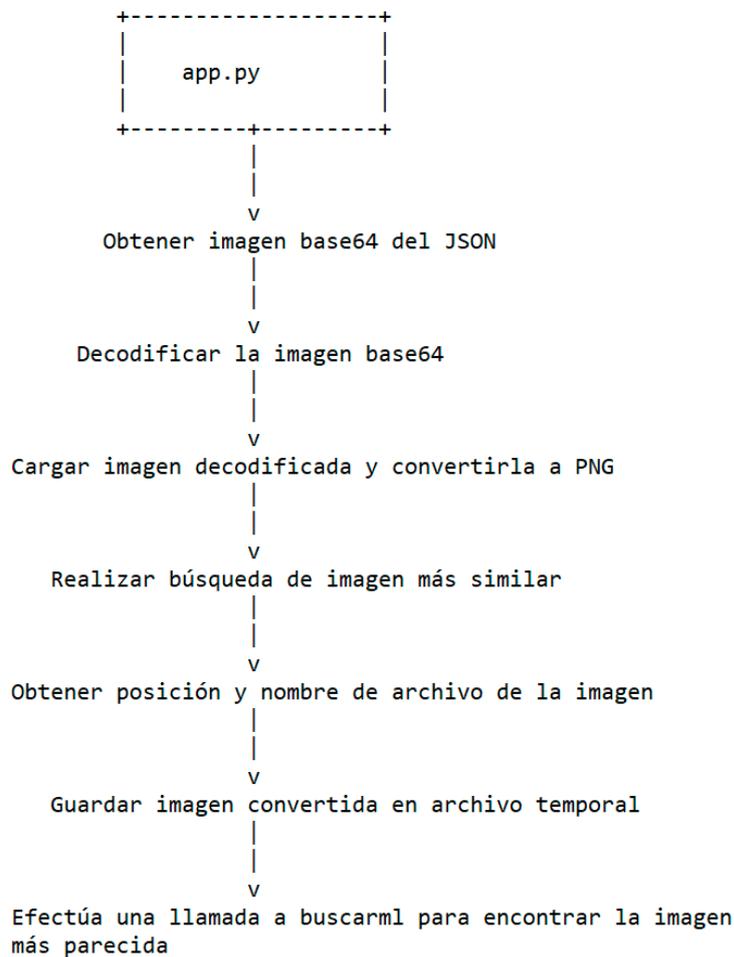


Figura 24: Esquema Backend

2.4.3. Extracción de Características

Debido a que la implementación de modelos de *machine learning* queda fuera del contenido del grado multimedia no se entrará en detalle debido a que se está realizando una publicación paralela con aspectos relacionados con la precisión y arquitectura del modelo, sin embargo, se describe el algoritmo en alta.

En el pipeline de nuestro trabajo la imagen tomada por la cámara se contrasta con un índice de búsqueda de vecinos aproximados (ANNOY) que nos devuelve cual es la imagen más próxima de nuestra posición de entre las sintéticas generadas.

Para este proceso previamente hemos congelado la última capa de un modelo *pre-entrenado* al que hemos cargado los pesos de *inception*, con este proceso separamos la clasificación del proceso de extracción de características, lo que nos permite ser más flexibles en tareas de agrupación o similitud visual. Sobre esta representación compacta de la imagen en un espacio vectorial podemos hacer búsqueda inversa *reverse search*. Los vectores de características nos aseguran que en un *dataset* con imágenes semejantes estas están próximas en el espacio vectorial.

Cuando se recibe la imagen de consulta, esta tiene que seguir el mismo proceso que las imágenes sintéticas, se extraen las características con los mismos parámetros de creación de árboles y modelo que en la generación de los índices. Luego, se utiliza el índice previamente creado para buscar las imágenes más similares a la imagen de consulta. La imagen más similar se determina **mediante la distancia euclidiana entre las características de la imagen de consulta y las características de las imágenes en el índice**. Finalmente, se devuelve el nombre del archivo de la imagen que haya alcanzado un mayor grado de probabilidad en la similitud.

Este nombre contiene la información de la cámara y la rotación. Se extraen estos datos: el numero de la cámara y rotación teniendo en cuenta la incremental de 15 grados en cada toma de rotación.

Las partes más importantes de este proceso son:

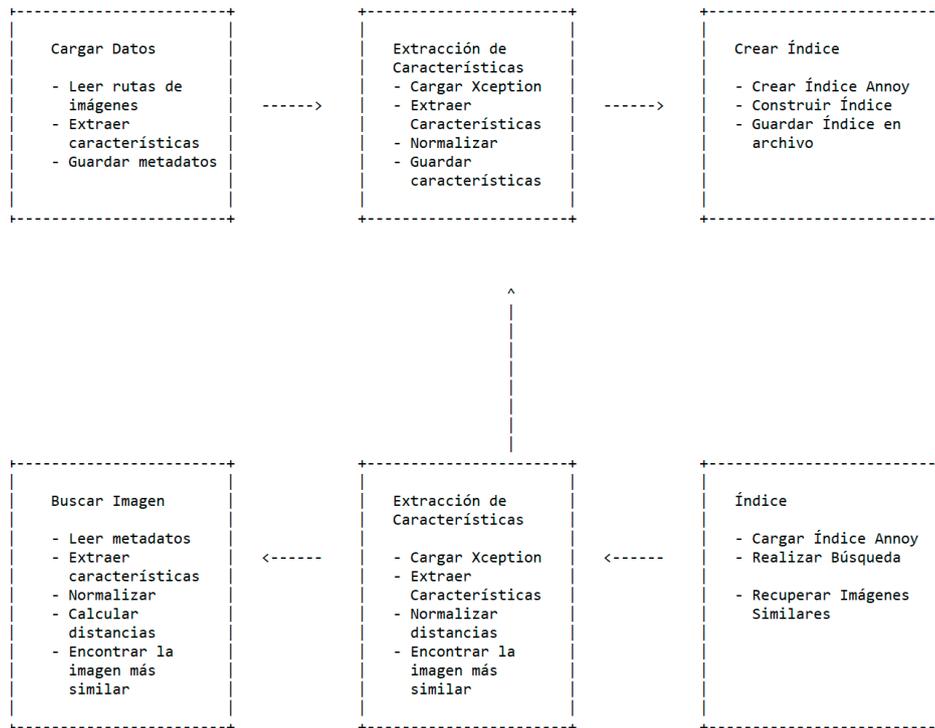


Figura 25: Esquema extracción características.

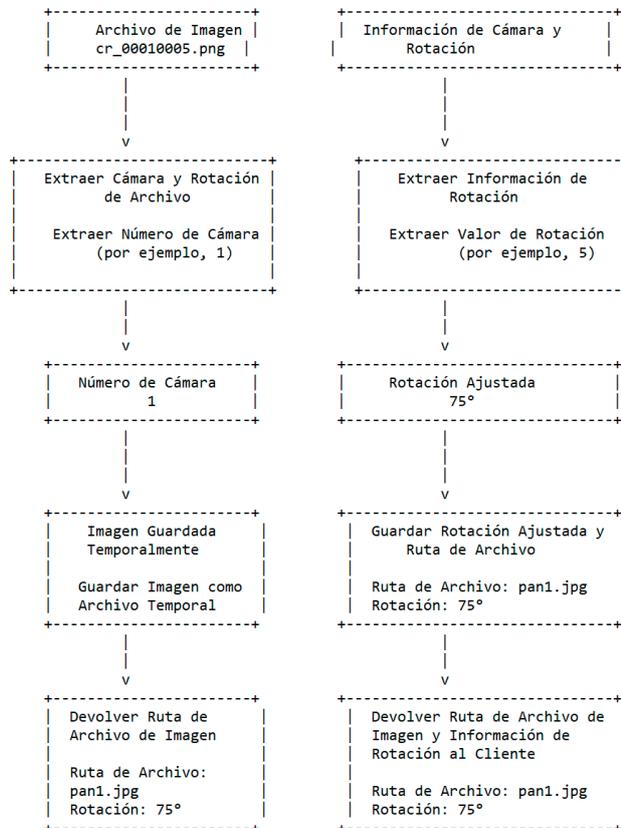


Figura 26: Esquema extraer cámara y grados

2.4.4. Respuesta del servidor

Flask y *Jinja* desempeñan un papel importante en este proceso. *Flask* es el *framework* de desarrollo web utilizado para manejar las rutas y las solicitudes del cliente, mientras que *Jinja* es un motor de plantillas que permite generar contenido dinámico en las páginas HTML. En este caso, *Flask* y *Jinja* se utilizan para renderizar y mostrar la página `display_image.html` con los parámetros adecuados.

Se accede a la ruta `/display_image.html`, se solicita la imagen correspondiente desde un contenedor `Blob`. Se crea un objeto de archivo en memoria y se transmite la imagen al cliente.

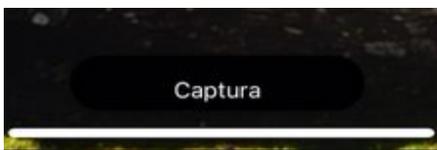
En este bloque de código, después de recibir una respuesta exitosa del servidor (estado de respuesta 200), se procesa la respuesta JSON. Se extraen la ruta de la imagen más similar (*temp_file*) y la información de posición (*position_name*).

Luego, se construye una URL para redirigir al usuario a la página `display_image.html` que muestra la imagen y posiciona la cámara virtual en la panorámica. Los parámetros de la imagen y la posición se codifican en la URL utilizando *encodeURIComponent* para asegurar que los caracteres especiales sean tratados correctamente.

2.5. Interfaz

La interfaz es **sencilla y minimalista**, la finalidad es que el usuario pueda tener una experiencia interactiva fluida.

El sistema consta de un botón de captura de imagen.



Un elemento animado para informar sobre el proceso de carga.



Dos botones: VR y para regresar a la pantalla principal con una cámara y un *affordance* (icono cámara para regresar al inicio).

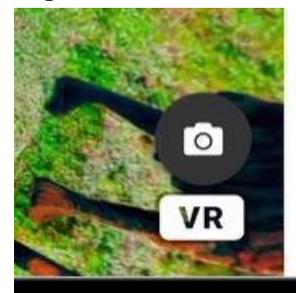


Figura 27: Botones interfaz

3. CONCLUSIONES FINALES

Este trabajo me ha permitido implementar muchas de las técnicas que he aprendido a lo largo del grado, siendo este uno de sus puntos fuertes, destacaría además aspectos como:

- **Originalidad y simplicidad del lado del cliente:** El enfoque se centró en brindar una experiencia fácil para el usuario final en el manejo de una interfaz intuitiva para una **interacción fluida** sin necesidad de conocimientos avanzados ni curva de aprendizaje.
- **Confiabilidad basada en inteligencia artificial:** La confianza en los resultados se basa en el poder de reconocimiento y similitud proporcionado por los algoritmos de inteligencia artificial. El método de reconocimiento de imágenes es preciso y potencialmente confiable para un entorno más amplio.
- **Invisibilidad de la tecnología:** El sistema se enfoca en **ocultar la complejidad tecnológica** al usuario final. Los detalles técnicos y la implementación de la inteligencia artificial se gestionan en el servidor, permitiendo que el cliente se enfoque en la experiencia visual y resultados obtenidos.
- **Eficiencia energética y consumo optimizado:** Hemos cuidado la eficiencia energética como un aspecto fundamental en el diseño del sistema. La elección de **un servidor eco** de *Heroku* garantiza un consumo energético optimizado, ya que **permanece en estado de espera (standby)** hasta que se recibe una solicitud. Esto contribuye a la reducción de la huella de carbono y al uso responsable de los recursos.
- **Servicio en la nube:** La solución se beneficia de la **escalabilidad y disponibilidad** proporcionadas por la nube. El servidor en *Heroku* permite manejar múltiples solicitudes y proporciona un tiempo de actividad confiable para garantizar la disponibilidad del servicio en todo momento.

En cuanto a las áreas de mejora crítica, es importante destacar que el método propuesto involucra un proceso laborioso y requiere una preparación exhaustiva, desde la creación de la fotogrametría a los renderizados de todo el entorno virtual. Sin embargo, en futuros trabajos se pueden explorar opciones que utilizan inteligencia artificial (IA) para interpolar posiciones intermedias entre imágenes, con tecnologías como NERF que pueden ofrecer soluciones más eficientes en términos de costes temporales de producción.

Además, para implementar este sistema de manera efectiva, se requiere una inversión mayor en planificación de servidores para obtener una capacidad de procesamiento de VRAM más alta. En el prototipo, se utilizó el plan ECO de *Heroku*, que tiene un costo de 5 euros mensuales. Sin embargo, se podría mejorar el rendimiento de la aplicación al optar por planes superiores que tienen un costo a partir de 20 euros mensuales. Esta inversión adicional en infraestructura permitiría manejar cargas de trabajo más demandantes y optimizar el rendimiento general del sistema.

4. PLANIFICACIÓN

PEC1 Definición y planificación del Trabajo final 19/03/2023

Investigación sobre la propuesta	5 d	
<i>Front-End</i> Acceso Cámara Mobile App	7 d	
<i>Backend</i> Montaje Servidor	4 d	
Redacción de la propuesta	4 d	

PEC2 Estado del arte o análisis de mercado del proyecto 16/04/2023

Análisis del mercado	5 d	
Estado del Arte	7 d	
Conversión y extracción de características	3 d	
Selección del entorno arqueológico	3 d	
Trabajo de campo: Digitalización Fotogramétrica	1 d	
Creación del modelo digital (generación, limpieza, texturizado)	2 d	
<i>Batch</i> renderizado para <i>dataset</i>	15 d	
<i>Feedback</i> Tutor	3 d	
Redacción TFR	3 d	

PEC 3 Diseño e implementación del Trabajo 14/05/2023

Implementación producto	47 d	
Diseño gráfico <i>interface</i>	3 d	
Prototipos Lo-Fi aplicación	3 d	
Prototipos Hi-Fi aplicación	3 d	
Reconstrucción Virtual	10 d	

Montaje Escenario preparación algoritmo cámaras	2 d	
Batch Renderizado Reconstrucción 360.	12 d	
Extracción características generación modelo	2 d	
Implementación sustitución RV en <i>frontend</i>	2 d	
Algoritmo para la predicción <i>neighbour nearest</i>	3 d	
<i>Feedback</i> tutor semanal	6 d	
Redacción y análisis de resultados	7 d	

PEC 4
Redacción Memoria (1a Entrega)

04/06/2023

Redacción	12 d	
Pruebas	8 d	
Presentación TFR	4 d	
Correcciones <i>feedback</i> Tutor	4 d	

PEC 5
Redacción Memoria (Entrega Final)

10/06/2023

Redacción	4 d	
Presentación TFR	3 d	
Grabación edición Video	3 d	
<i>Feedback</i> Tutor	4 d	

Preparación Defensa del proyecto

12/06/2023

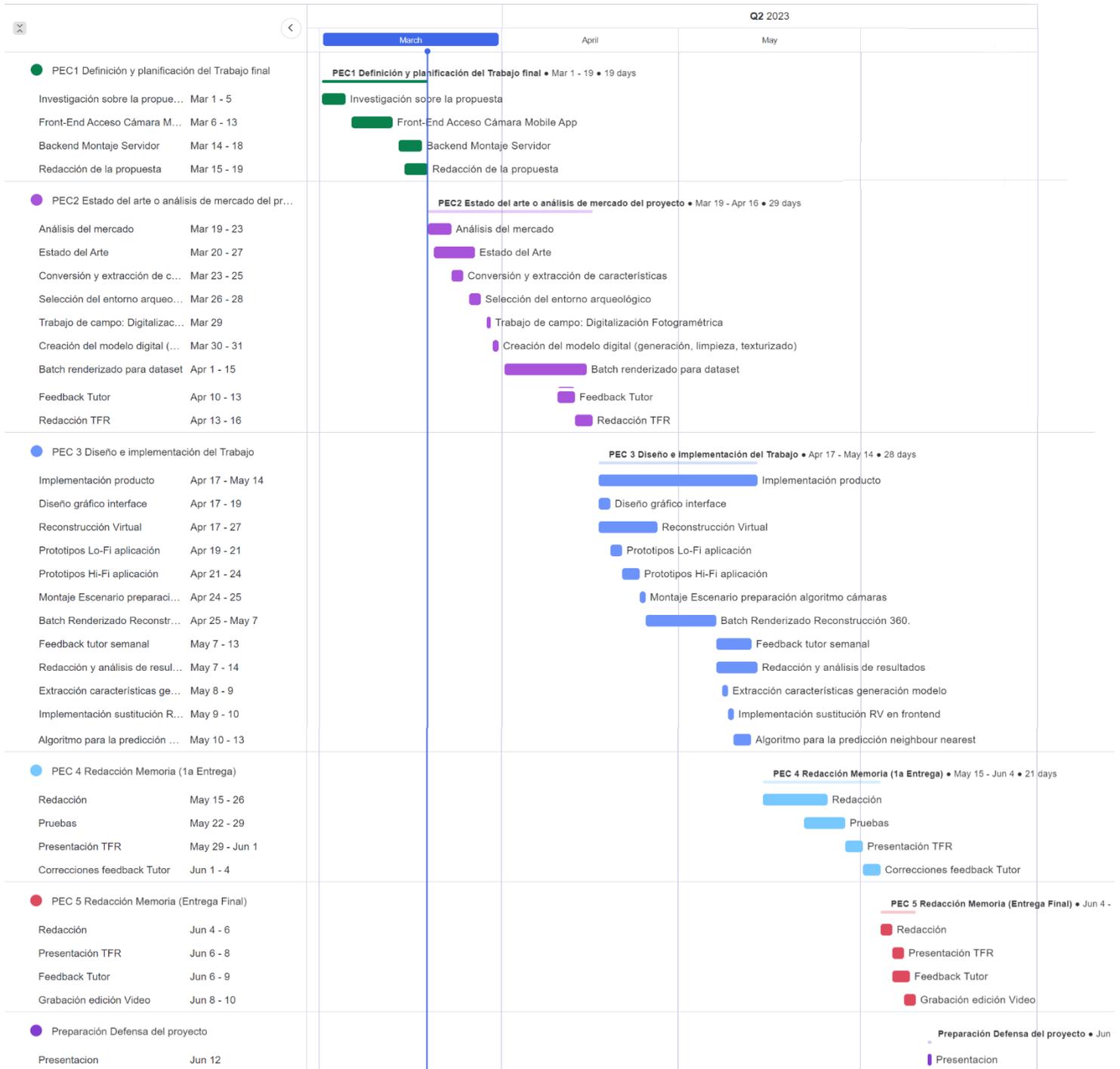
Defensa Virtual

14/06/2023

Desglose de actividades

			
Documentación Redacción	Programación Multimedia	<i>Feedback</i> Tutor	Entregas

5. Diagrama de Gantt



BIBLIOGRAFIA

- [1] C. Masriera, "Presentación del patrimonio arqueológico: ruinas versus reconstrucciones, ¿Qué entiende más el público?," *Didáctica las Ciencias Soc. Geogr. e Hist.*, vol. 57, pp. 39–51, 2008.
- [2] P. Reilly, "Towards a virtual archaeology," in *Computer Applications in Archaeology*, 1990, pp. 133–139.
- [3] A. Almagro Gorbea, *De la Fotogrametría a la Infografía: un proceso informatizado de documentación*. 2003. [Online]. Available: <http://hdl.handle.net/10261/19816>
- [4] V. M. Bendicho and A. Grande, "The principles of the Seville Charter," *XXIIRD CIPA Symp.*, pp. 2–6, 2011.
- [5] Google, "Implementing 'magic window' in Unity," 2019. <https://developers.google.com/vr/develop/unity/guides/magic-window>
- [6] C. J. Balck, "User Interaction in Inflight Entertainment Map Application," Faculty of engineering LTH Lund University, 2019. [Online]. Available: <https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8989614&fileId=8989620>
- [7] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, 2006, doi: 10.1109/MRA.2006.1638022.
- [8] H. L. Dibble, C. W. Marean, and S. P. McPherron, "The use of barcodes in excavation projects: examples from Mossel Bay (South Africa) and Roc de Marsal (France)," *SAA Archaeol. Rec.*, vol. 7, no. 1, pp. 33–38, 2007.
- [9] K. N. Hernández, "San Lorenzo del Escorial gana el Premio a la Mejor App Turística Nacional." <https://www.gndiario.com/san-lorenzo-del-escorial-gana-el-premio-la-mejor-app-turistica-nacional>
- [10] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM Algorithm for Robust Localization and Mapping," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 24–29. doi: 10.1109/ROBOT.2005.1570091.
- [11] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking," *J. Sensors*, vol. 2021, p. 2054828, 2021, doi: 10.1155/2021/2054828.
- [12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015, doi: 10.1109/TRO.2015.2463671.
- [13] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *2017 IEEE international conference on robotics and automation (ICRA)*, 2017, pp. 4503–4508.
- [14] R. Newcombe, S. Lovegrove, and A. Davison, *DTAM: Dense tracking and mapping in real-time*. 2011. doi: 10.1109/ICCV.2011.6126513.
- [15] S. G. Koul, Anirudh and M. Kasam, *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & Tensorflow*. O'Reilly Media, INC, 2019.
- [16] A. Kiran *et al.*, "Reverse Image Search Using Deep Unsupervised Generative Learning and Deep Convolutional Neural Network," *Appl. Sci.*, vol. 12, no. 10, 2022,

- doi: 10.3390/app12104943.
- [17] R. Ranjan, S. Gupta, and K. S. Venkatesh, "Image retrieval using dictionary similarity measure," *Signal, Image Video Process.*, vol. 13, no. 2, pp. 313–320, 2019.
 - [18] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.
 - [19] W. Niblack, "Storage and retrieval for image and video databases," in *Storage and Retrieval for Image and Video Databases*, 1993, vol. 1908.
 - [20] S. Das, S. Garg, and G. Sahoo, "Comparison of content based image retrieval systems using wavelet and curvelet transform," *Int. J. Multimed. \& Its Appl.*, vol. 4, no. 4, p. 137, 2012.
 - [21] R. Liu, Q. Yu, and S. Yu, "Unsupervised Sketch-to-Photo Synthesis," Sep. 2019, Accessed: Jun. 11, 2020. [Online]. Available: <http://arxiv.org/abs/1909.08313>
 - [22] S. K. Addagarla and A. Amalanathan, "e-SimNet: A visual similar product recommender system for E-commerce," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 563–570, 2021.
 - [23] P. Desai, J. Pujari, C. Sujatha, A. Kamble, and A. Kambli, "Hybrid Approach for Content-Based Image Retrieval using VGG16 Layered Architecture and SVM: An Application of Deep Learning," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–9, 2021.
 - [24] S. B. and A. Fulgosi, "The influence of pattern similarity and transfer learning upon the training of a base perceptron B2," in *Proceedings of Symposium Informatica*, 1976, pp. 3-12 1–5.
 - [25] S. Bozinovski, ""Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, pp. 291–302, 2020.
 - [26] W. Li, Y. Zhang, Y. Sun, W. Wang, W. Zhang, and X. Lin, "Approximate Nearest Neighbor Search on High Dimensional Data --- Experiments, Analyses, and Improvement (v1.0)." arXiv, 2016. doi: 10.48550/ARXIV.1610.02455.